Incorporating Risk

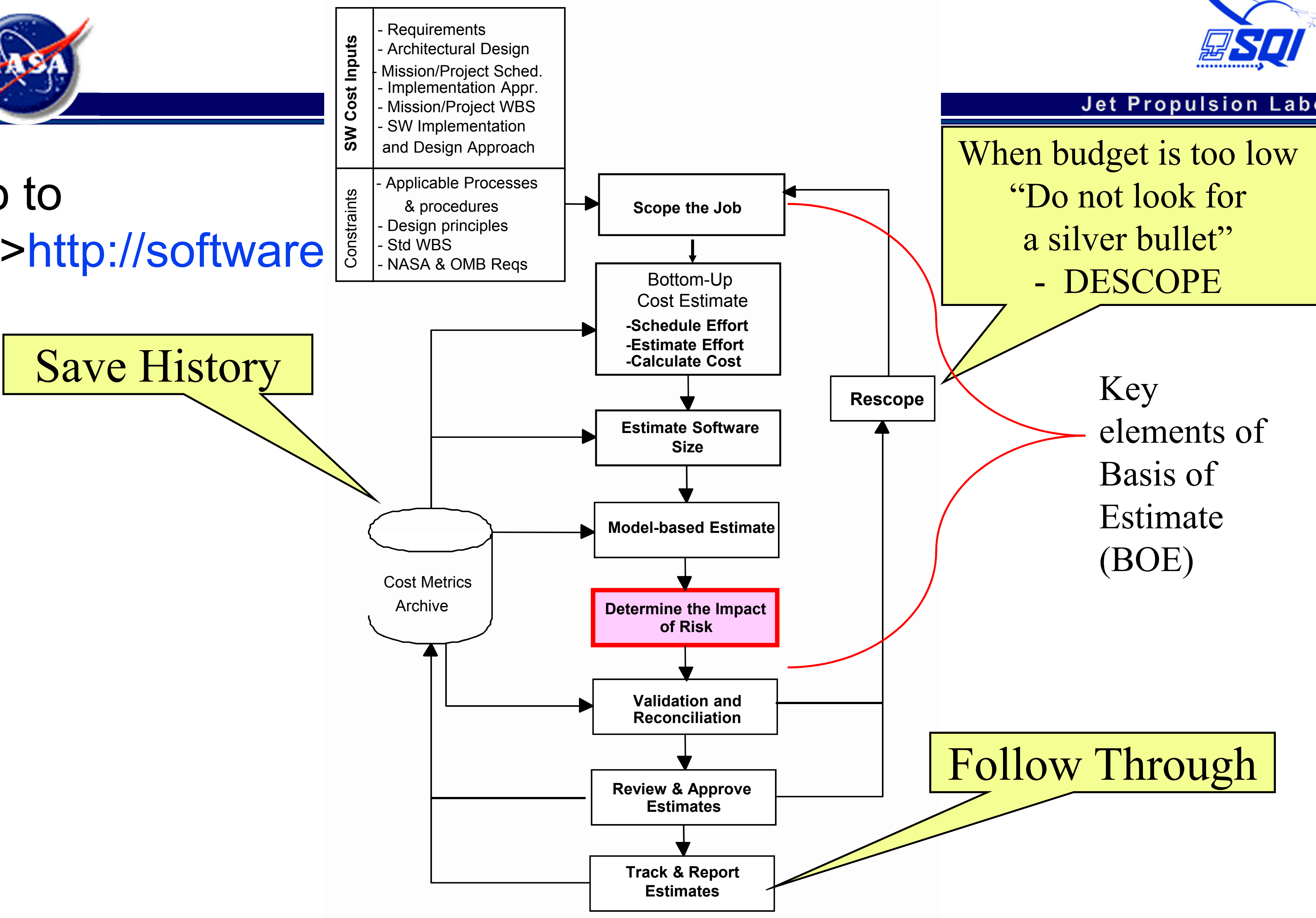
Presented by:

Jairus Hihn

This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Go to

>http://software



Incorporating Risk

- The purpose of this step is to identify common software risks, to assess their impact on the cost estimate, and to make revisions to the estimate based on these impacts
- Risk can be estimated and analyzed in various ways
  - Risk Matrices
  - Expected Risk (Likelihood * Impact)
  - Monte Carlo techniques to capture distributions
  - Analyses tools such as ARRT (Advanced Risk Reduction Tool) & DDP (Defect Detection Process)
- List known <u>major</u> risks.
  - List should be approximately 7+/- 2 items
- Include indicator of likelihood and impact
  - Can be categorical (Low, Medium, High) or numerical
  - Especially try to determine the cost impact

Background

- A risk is an event that has the potential to cause significant impact on technical, cost and/or schedule performance.
- This presentation is about cost risk identification and estimation which is only a part of risk management
- Risk management is an aspect of overall management and includes
  - cost risk
  - schedule risk (integrated network schedules & critical path)
  - technical risk (good at this but need to map into cost & schedule risk)
- Risk management should be conducted consistently with the NASA risk management process
  - NASA/SP-2010-576 (VER. 1.0), NASA SPECIAL PUBLICATION: NASA RISK-INFORMED DECISION MAKING HANDBOOK
  - NPR 8000.4A. Agency Risk Management Requirements

Cost Uncertainty: Incorporating Uncertainty in Your Estimates

Jet Propulsion Laborat

- There are two main recommended techniques for addressing risk and uncertainty

1. Construct the risk matrix
   - The NASA recommended risk matrix requires estimates of likelihood of occurrence and impact by categories
2. Make all estimates as distributions and use Monte Carlo techniques to combine the estimated elements of the project.
   - This is what you learned in the sizing and model lectures
   - Monte Carlo approaches require specifying the parameters of a distribution such as
     – low and a high for a Uniform
     – low, most likely, and high for a Triangular distribution

Cost Risk Concepts & Definitions – Sources of Cost Uncertainty

Source	How Addressed
Known Unknowns	Risk Lists Risk Assessment
Unknown Unknowns	Design Principle Reserve %

Cost Risk Concepts, Definitions – Guidelines

- Formal cost risk identifies known unknowns
- Percentage reserve guidelines cover the unknown unknowns
- Risk approach should be simple to understand, use, and track
- Use cost risk estimate to identify reasonable margin
- Flow up to project with cost estimate
- Risk drivers are those events with high probability of occurrence and significant consequence
- Assessing risk at too low a level does not provide any added value

Risk Identification – Generating the Risk List

- As you generate the software risk list, think about
  - What WBS elements are affected
  - When it would occur
  - Likelihood of occurrence
  - Impact
  - What it would cost to fix it
- Start with project significant risk list (SRL) and common risks
- Develop software level significant risk list (SSRL)
  - Link to specific events for specific task
  - Link to specific WBS elements
  - Link directly to design
  - Identify finite number of “big ticket” items or main risk drivers

Common Causes of Effort Growth

Historically, there is a pattern of being overly optimistic in setting budgets by not taking sufficient account for:

- Changes and increases in scope
  - Concurrent hardware development
  - Inability to scope flight software due to inadequate project definition
  - Software is used for risk mitigation, but never planned for up front.
  - Software is the system complexity sponge
- Testbed and SoftSim availability and maturity
- Optimistic software inheritance assumptions
- Anything New
  - Technology
    - Autonomy
    - Precision landing
    - Hazard avoidance
  - Design
  - Language
  - Tools
  - Development environment
  - Processes
  - Customer or sponsor

[These items are based upon causes of cost growth observed at JPL.]

Risk Identification – Identifying Main Risk Items

- Systematically go through WBS and identify risk items
- Remember to consider design, system level risk lists, risk check lists

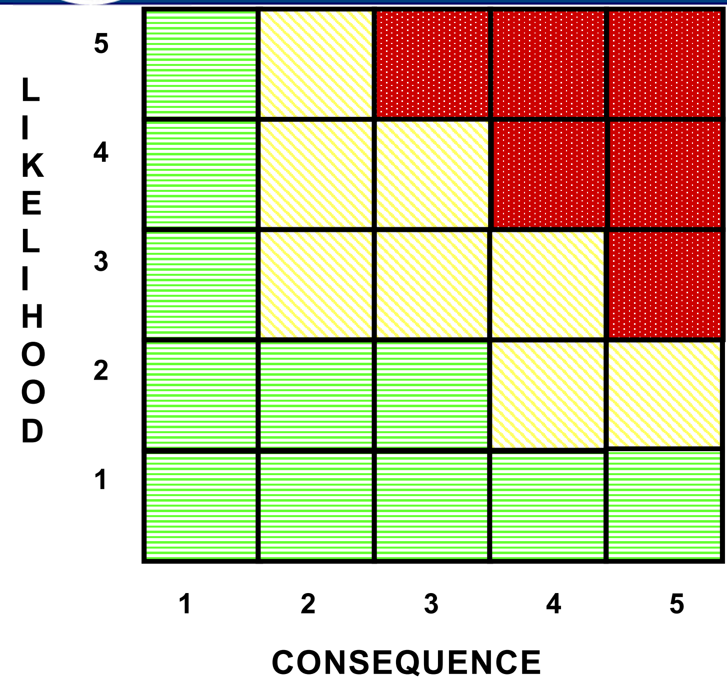
	WBS Element	Risk Item
Spacecraft Flight Software		
	Software Management	
	Software Systems Engineering	Technical margins below Flight Practice Desing Principles
	GN&C	Autonomy
	CT&DM	
	Sequencing	
	Engineering Appplications	
	Payload Accomodation	Instrument delivery could be late
	Fault Protection	Current implementation assumes siginificant SW inheriteance
	Software Development Testbed	
	Software Integration & Test	Schedule crunch / additional FTEs

Risk Identification
Constructing the Risk List

- Construct risk list from identified risk items
- Document basic reasons, associated issues, assumptions for identifying each risk item

Risk Item	Description	Mitigation Action
Autonomy	Existing planner has had some reliability issues that are not fully understood.	Start aggressive prototyping activity immediately to fully identify the issues. In operations could reduce scope of autonomous operations.
SW Inheritance Assumptions	Inherited software does not perform as expected.	Current level of inheritance is only 10-15%. Will have to write code from scratch. Can hold inheritance review right after PDR to better determine how much code can really expect to inherit. Set go-no-go decision point before CDR so can start planning for new code development as early as possible. Can descope to reduce impact.
Low CPU Margins	If CPU margins are too low then do not have sufficient flexibility to handle failures. This can highly constrain design and drives up cost and cost variance significantly.	Oversize system (increase cost) and manage technical margins very carefully. Also carry larger reserves.
Insufficient I&T Schedule	Preceding activities typically over run their schedule putting schedule pressure on I&T. Creates pressure to descope testing activities.	Budget for running multiple shifts.
Late instrument delivery	University XXX has delivered late the last two missions and has not always delivered to specificiations.	Hold 1 month fully funded schedule reserve to cover possible code changes and extra testing activities.

NASA Risk Matrix and Criteria
Project Level



Criticality

- **Implement new process(es) or change baseline plan(s)**
- **Aggressively manage; consider alternative process**
- **Monitor**

Risk Assessment Criteria for Consequence and Likelihood

Risk Type (either or both may apply to each risk)
Mission Risk
Implementation Risk

Consequence of Occurrence

Level	Mission Risk Level Definitions
5	Mission failure
4	Significant reduction in mission return
3	Moderate reduction in mission return
2	Small reduction in mission return
1	Minimal (or no) impact to mission

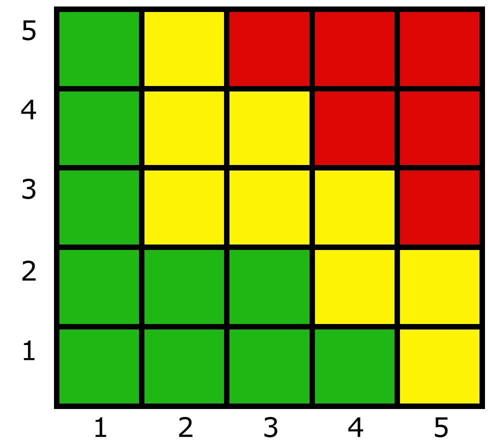
Level	Implementation Risk Level Definition
5	Overrun budget and contingency, cannot meet launch with current resources
4	Consume all contingency, budget or schedule
3	Significant reduction in contingency or launch slack
2	Small reduction in contingency or launch slack
1	Minimal reduction in contingency or launch slack

Likelihood of Occurrence

Level	Likelihood	Level Definition
5	Very High	>70%, almost certain
4	High	>50%, More likely than not
3	Moderate	>30%, Significant likelihood
2	Low	>1%, Unlikely
1	Very Low	<1%, Very unlikely

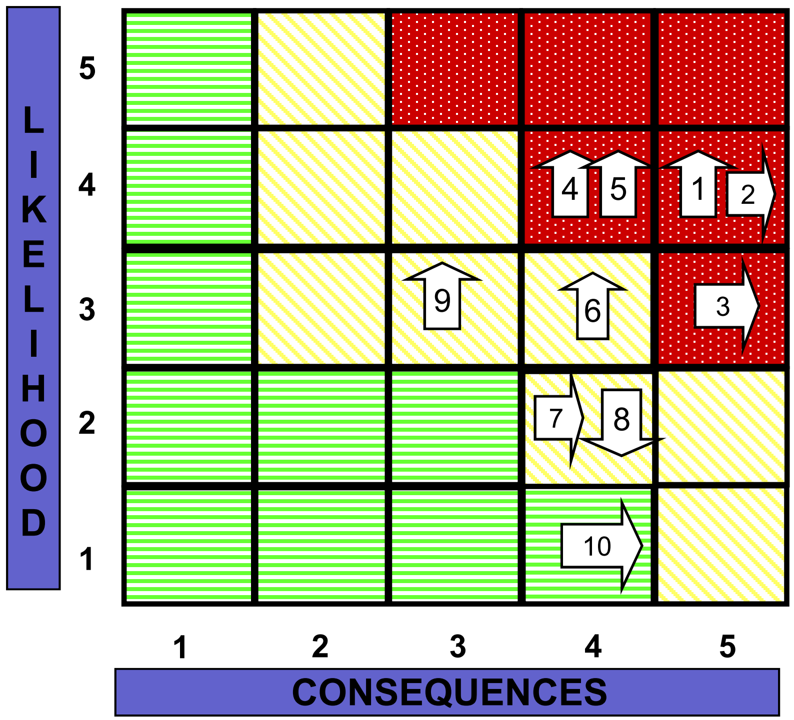
Recommended Risk Matrix and Criteria Software

Likelihood	
1	Very low - Very unlikely
2	Low - Unlikely
3	Moderate - Significant likelihood
4	High - More likely than not
5	Very high - Almost certain



Consequence	
1	Minimal or no impact to deliverable set of FSW
2	Small reduction in deliverable set of FSW
3	Serious reduction in deliverable set of FSW
4	Cannot meet a functional minimum set of FSW
5	FSW products fatally flawed or unusable by Missions

Top Risk List & Risk Matrix Example



Criticality	L x C Trend	Approach
High	⇩ Decreasing (Improving)	M - Mitigate
Med	⇧ Increasing (Worsening)	W - Watch
Low	⇨ Unchanged	A - Accept
	☐ New Since Last Period	R - Research

SW Design Phase

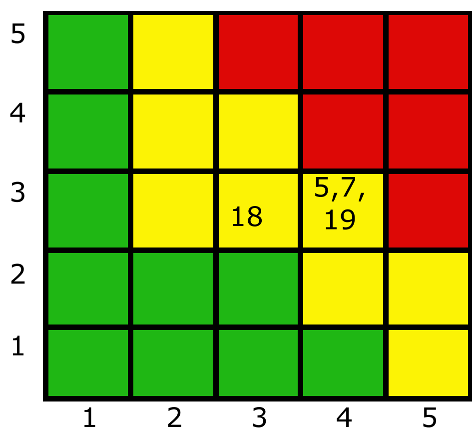
Rank	Approach – *Title* – Description
1	M – *Requirements Management* - Baseline has not been possible due to too many changes being requested.
2	M - *Technical Expertise* - Have not been able to find more Java programmers with experience. Got some resumes to go through.
3	M - *Aggressive Schedule* - There is no time to do all scheduled reviews
4	M – *Interface Design* - The interface design with XXX-hardware component is not completed due to limited access to the engineers.
5	R – *Software Testbeds* - All the requirements are not in for the tests beds. Testing will be delay if test bed not ready by XXX.
6	R – *Test procedures* - The procedures are not completed.
7	A – *Software Reliability* - The reused software is not as reliable as expected.
8	W – *Contractors Deliveries* - The second delivery of the SW is very critical for the project..
9	M – *SAQ Support* - No funds for an independent SQA resource.
10	M – *Manager's Visibility* - Inadequate, incorrect, or inefficient processes being used.

[Risks are identified and trended from last review to current review]

Project Example
Top/Changed FSW Risks

Jet Propulsion Laborat

Item #	Category: Name	Description	Risk consequence to MSAP	Consequence Rating	Likelihood Rating	Possible Mitigations	Status/Date	Mitigation Taken / date
5	Development: FSW Requirements Risk	FSW implementation is proceeding while requirements have yet to be fully defined, possible risk to current work does not meet requirements.	Rescope or rework to meet ultimate requirements.	4	3	Increased systems involvement through meeting attendance and FDD reviews.	Mitigation ongoing 10/10/06	Adding "rework" slots to schedule to close gaps between implementation and requirements. 12/12/06
7	Resources: Talent availability/attrition	Talent is unavailable or siphoned off, leaving project without capabilities required to meet schedule.	Unable to meet functionality on schedule. Schedule stretch-out.	4	3	1) Usually little recourse. Work agreements with Line.	Mitigation taken 6/22/05 Watching 8/26/05 Closed 11/4/05 Reopened 3/6/06	1) Concentrating on full-time people 2) Physical co-location in T1718 6/22/05 Staff is set; will work to defend when people are being stolen. 12/12/06
18	Devlopment: Schedule impacts due to testbed problems.	Module development is delayed due to testbed availability and/or problems. Avionics hardware, GSE, and test software are contributors.	Schedule erosion. Late deliveries and/or reduced capability as functionality is descoped.	3	3	Test more on FuncSim and BitSim. Begin early work on subsequent modules	Mitigation ongoing 10/10/06	Still too much focus on h/w to s/w debugging 5/9/06 Focus of testing shifting to sim environments. 12/12/06
19	Devlopment: Schedule impacts due to required changes.	Maintenance, change, or enhancement work is unknown and not separately scheduled.	Schedule erosion. Functional capabilities could be slipped to later builds or off of the committed list.	4	3	Control changes using Software Change Board as a control point. Including rework effort in schedule estimates. Improve code review activities.	Watching: MSL inheritance reports 10/10/06	Improving coding standards and enforcement 12/12/06



5 - Unchanged; fitting rework with new work

7 - Unchanged; will continue to be vigilant

18 - Unchanged; more focus on sim environments for testing

19 - Unchanged; agressively working changes/problems from MSL

Project Example
Behind the scenes …

Item #	Category: Name	Description	Risk consequence to MSAP	Consequence Rating	Likelihood Rating	Possible Mitigations	Status/Date	Mitigation Taken / date
4	Resources: FSW Capability	Unable to meet user requirements with FSW that is delivered due to resources available	Unable to meet user expectation	4	2	Produce a prioritized FSW capability list to make sure Users are aware of the risk areas	Mitigation ongoing 10/10/06	Build content has been prioritized. Upscope/descope lists are kept current. Working with User Missions to coordinate Build and Phase contents. Phase 1 delivery rescoped at FSW CDR. Tracking earlyand late dates to see range. 11/4/05
5	Development: FSW Requirements Risk	FSW implementation is proceeding while requirements have yet to be fully defined, possible risk to current work does not meet requirements.	Rescope or rework to meet ultimate requirements.	4	3	Increased systems involvement through meeting attendance and FDD reviews.	Mitigation ongoing 10/10/06	Adding "rework" slots to schedule to close gaps between implementation and requirements. 12/12/06
6	Development: FSW Adaptability	Adaptation process is not as streamlined as users expect. System is not adaptable …	Unable to meet adaptability requirement, more user mission support from project required for adaptation process.	3	2	1) Include adaptability as a review item. 2) Perform "trial adaptations". 3) Increased support to user missions for adaptation.	Mitigation taken 6/22/05 Watching 5/9/06	1) Adaptability added as a design/review item. 6/22/05 2) Note potential adaptability upscopes, document adaptation features in SDDs
7	Resources: Talent availability/attrition	Talent is unavailable or siphoned off, leaving project without capabilities required to meet schedule.	Unable to meet functionality on schedule. Schedule stretch-out.	4	3	1) Usually little recourse. Work agreements with Line.	Mitigation taken 6/22/05 Watching 8/26/05 Closed 11/4/05 Reopened 3/6/06	1) Concentrating on full-time people 2) Physical co-location in T1718 6/22/05 Staff is set; will work to defend when people are being stolen. 12/12/06
9	Technology: FSW Performance	Unable to meet the performance requirements on cpu performance or memory usage.	Unable to meet user mission expectations	4	2	Develop performance analyses to characterize actual performance. Apply "truth in advertising" rule.	Mitigation ongoing 10/10/06	Analyses being developed. Will be included in SW I&T testing.
11	Planning: Poor module effort estimation	Module development effort estimation has been done only by "rule of thumb estimates" which could have large error bars. Overall estimation was done using cost models.	Schedule erosion. Late deliveries and/or reduced capability.	3	3	Capture development history and use for corroborating module estimates for future builds	Mitigation ongoing 10/10/06	Effort, size, complexity, and test extent being kept for developed modules. Being used for estimation corroboration. Trying to characterize "testbed issues" for future reference.
18	Devlopment: Schedule impacts due to testbed problems.	Module development is delayed due to testbed availability and/or problems. Avionics hardware, GSE, and test software are contributors.	Schedule erosion. Late deliveries and/or reduced capability as functionality is descoped.	3	3	Test more on FuncSim and BitSim. Begin early work on subsequent modules	Mitigation ongoing 10/10/06	Still too much focus on h/w to s/w debugging 5/9/06 Focus of testing shifting to sim environments. 12/12/06
19	Devlopment: Schedule impacts due to required changes.	Maintenance, change, or enhancement work is unknown and not separately scheduled.	Schedule erosion. Functional capabilities could be slipped to later builds or off of the committed list.	4	3	Control changes using Software Change Board as a control point. Including rework effort in schedule estimates. Improve code review activities.	Watching: MSL inheritance reports 10/10/06	Improving coding standards and enforcement 12/12/06

Risk Matrix Analysis (1)

- Has the Team thought through potential threats?
  - Have all the risks been identified (Is anything missing?)
  - Do the assessments make sense (Does it pass the “laugh test”?)
    - Impact
    - Likelihood
  - Drive high likelihood risks into budget (5’s)
  - Identify major risk drivers and determine if redesign can lower risk in these areas

Risk Matrix Analysis (2)

- Identify all red risk items
- Especially determine risk mitigation strategies for these risks and baseline the mitigation costs into relevant WBS element if mitigation is cheaper then holding cost margin
  - This is where ARRT/DDP can be applied
- Else add cost margin to relevant cost elements for identified risk items
- Flow up uncovered SSRL risk items to project
- If budget gets pushed down by manager who does not really understand software, then use risk list and matrix to show impact on risk
  - Also remember: DESCOPE!

Bottom-Up Cost Risk Estimate

- Derive Cost Risk Estimate by
  - eliciting, for each WBS element, the worst case, most likely and best case cost,
  - then integrating with Monte Carlo methods

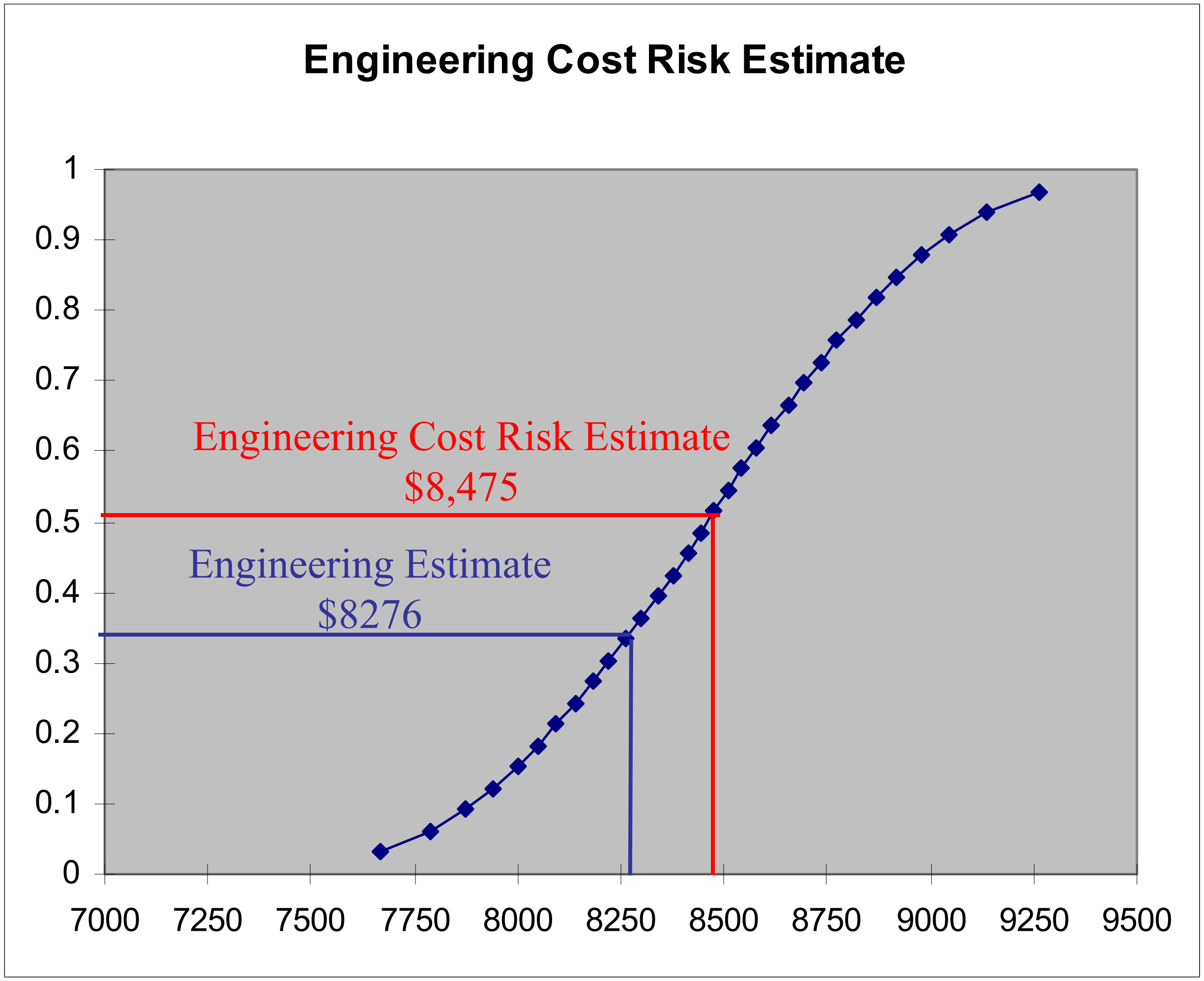
Bottom-Up Cost Risk Estimate

- Develop cost risk methodology using engineering cost estimate
- Develop risk distributions
  - Cost risk assessment by WBS (cost, schedule, technical & programmatic)
    - Performed at the system, subsystem or component level
  - Determine probability distribution for each WBS element
  - Triangular Distribution: Low, Most Likely (Budget), High
  - Log-normal Distribution: Pessimistic cost either as a Cost or a % of budget
- Run Monte Carlo simulation to combine risk distributions to produce total project cost probability distribution
- Involves subjective expert judgment and/or engineering assessment

Bottom-Up Cost Risk Estimate – Alternate Approach Inputs

	WBS Element		Low -10% ($K FY03)	Budget ($K FY03)	High -90% ($K FY03)		Risk Item
Power S/S							
		Software Management	800	899	899	866	
		Software Systems Engineering	750	950	1321	1007	Technical margins below Flight Practice Desing Principles
		GN&C	1850	2761	3367	2659	Autonomy
		CT&DM	1350	1492	1959	1600	
		Sequencing	500	543	600	548	
		Engineering Appplications	275	298	350	308	
		Payload Accomodation	200	275	300	258	Instrument delivery could be late
		Fault Protection	750	858	1206	938	SW inheritance
		Software Development Testbed	50	75	100	75	
		Software Integration & Test	100	125	175	133	Schedule crunch / additional FTEs
			6625	8276	10277	8393	

Bottom-Up Cost Risk Estimate



Wrap UP

- Estimating the cost risk enables the Manager or CogE to:
  - Identify reasonable margin
  - Identify when mitigation actions are needed
  - Be able to show quickly what is changing when budgets get pushed down. Lower budgets mean higher risk and decreased scope

Appendix

Software Cost Risk Drivers and Ratings

Risk Drivers	Software Cost Risk Driver Ratings	
	Nominal (Reduces Risk)	Extra High (Increases Risk)
Experience & Teaming	• Extensive software experience in the project office • Software staff included in early planning and design decisions • Integrated HW and SW teams	• Limited software experience in the project office • Software staff not included in early planning and design decisions • HW and SW teams are not integrated
Planning	• Appropriately detailed and reviewed Plan • All key parties provide input with time to get buy-in • Appropriate assignment of reserves • SW inheritance verified based on review and adequate support	• Lack of appropriate planning detail with insufficient review • Not all parties involved in plan development • Simplistic approach to reserve allocation • Optimistic non-verified assumptions especially with respect to software inheritance
Requirements & Design	• Solid system and SW architecture with clear rules for system partitioning • Integrated systems decisions based on both HW and SW criteria • SW Development process designed to allow for evolving requirements	• System and Software architecture not in place early with unclear descriptions of basis for HW & SW partitioning of functionality. • Systems decisions made without accounting for impact on software • Expect SW requirements to solidify late in the life-cycle
Staffing	• Expected turnover is low • Bring software staff on in timely fashion • Plan to keep software team in place through launch	• Expected turnover is high • Staff up software late in life-cycle • Plan to release software team before ATLO
Testing	• Multiple Test-beds identified as planned deliverables and scheduled for early completion. • Separate test team • Early development of test plan	• Insufficient Test-beds/simulators dedicated to SW and are not clearly identified as project deliverables • Plan to convert SW developers into test team late in life-cycle • Test documents not due till very late in the life-cycle
Tools	• CM and Test tools appropriate to project needs • Proven design tools	• No or limited capability CM and test analysis tools • Unproven design tools selected with limited time for analysis

Estimated Cost Impact of Risk Drivers

Risk Drivers	Estimated Cost Impact		
	High	Very High	Extra High
Experience & Teaming	1.02	1.05	1.08
Planning	1.10	1.17	1.25
Requirements & Design	1.05	1.13	1.20
Staffing	1.02	1.05	1.13
Testing	1.05	1.08	1.15
Tools	1.02	1.03	1.10
Maximum Expected Cost Impact	1.30	1.60	2.32

Rules-of-Thumb (1)

- JPL-Based "Rules-of-Thumb":
  - Software development costs typically overrun by 50% and can have an overrun greater than 100%.
  - On average, based on plans at PDR for DSMS upgrades, software cost overruns are 46% and schedules slip by 14%.
  - Based on 22 projects or upgrades at JPL, four out of five attempts to inherit major software code elements have failed
  - The six risk drivers, in the Tables 11 and 12 were identified based on a study of seven JPL missions that experienced significant cost growth [Hihn and Habib-agahi, 2000]

Rules-of-Thumb (2)

- "Rules-of-Thumb" from other Sources:
  - 55% of software projects exceed budget by at least 90%.
    - Software projects at large companies are not completed 91% of the time
    - Of the projects that are completed, only 42% of them have all the originally proposed features [Remer, 1998].
  - Historical cost estimates for NASA projects are underestimated by a factor of at least 2
    - The actual versus estimated cost ratio is from 2.1 to 2.5 [Remer, 1998]
  - Cost estimation accuracy using ratio estimating by phases without detailed engineering data gives an accuracy of –3% to +50%
    - Using flow diagram layouts, interface details, etc. gives an accuracy of –15% to +15%
    - Using well defined engineering data, and a complete set of requirements gives an accuracy of –5% to +15% [Remer, 1998]

Rules-of-Thumb (3)

- An accuracy rate of –10% to +10% requires that 7% of a rough order of magnitude budget and schedule be used to develop the plan and budget
  - Another way to look at this is to consider the percentage of total job calendar time required
  - When using existing technology, 8% of calendar/budget should be allocated to plan development
  - When high technology is used, then 18% of calendar/budget should be allocated to plan development [Remer, 1998]
- According to Boehm [Boehm, et. al., 2000], the impacts of certain risk drivers can be significantly higher than the JPL study:
  - Requirements volatility can increase cost by as much as 62%
  - Concurrent hardware platform development can increase cost by as much as 30%
  - Incorporating anything for the first time, such as new design methods, languages, tools, processes can increase cost by as much as 20%, and if there are multiple sources of newness, it can increase cost as much as 100%